Configurable Semi-Autonomic Animated Animal Characters in Interactive Virtual Reality Applications

Zhuming Ai[*] Mark A. Livingston[†]

Naval Research Laboratory

Abstract

In many virtual reality (VR) simulation and training applications, it is desirable and even critical to have computer controlled animal characters that can behave with a high degree of realism. The realism can be measured in two aspects. One is behavioral realism, or how real the characters act and respond to the commands and environments. The other aspect is visual realism.

Many efforts have been directed at animating human characters as well as other animal characters in applications such as interactive computer games. However, more research is still needed for realistic animation of animal characters due to the vast variety of animal species and different application purposes. In some of the environments for which security personnel are being trained, animals, such as dogs, are part of the training programs. So we need to include realistic behavior and visual representations of animal characters in our training applications.

This study is focused on animal behavior and animation for VR applications. A simple implementation of real-time animal animation method that is configurable is proposed to make it easy for user interaction. Synthetic animal characters are included in the system so that their behavior can be programmed and controlled digitally. A two stage state machine is used in the system. The top level state machine controls the animal behavior. The secondary state machine controls animation. Animation blending and procedure control are used to make the animation smooth.

Index Terms:

I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Virtual reality

I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

1 Introduction

In many virtual reality (VR) simulation and training applications, it is desirable and even critical to have computer controlled characters that can behave with a high degree of realism. In some of the environments for which security personnel are being trained, animals may be part of the training programs. So we need to include realistic behavior and visual representations of animal characters in our training applications. One such example is that dogs are used extensively in security and law enforcement environments, and there is a need to include realistic dogs in VR training and simulation applications.

Many efforts have been directed at animating human characters in applications such as interactive computer games and motion pictures.[14, 15] We have also seen realistically animated four-legged characters in movies. Realistic simulation of animals in video games can be displayed in real-time. Quadruped animation has made much progress in recent years.[17] There are many efforts on animal animation, especially research on articulated animal movement.[21]

For our purposes, we are more interested in domestic animals that are well-trained and behave in a more predicable manner. Our efforts are concentrated on how the animals are directed by users, but meanwhile react to the environment based on their training as well as instinct. Efforts have been made in our implementation to make such behaviors easily configurable.

The realism can be measured in two aspects. One is behavioral realism, or how real the characters act and respond to the commands and environments. The other aspect is visual realism.

1.1 Behavior Control using AI

Much research has been done on artificial intelligence (AI) for computer games[9], robotics[22], and VR simulation and training[6]. Behavioral models have been used to generate computer animation. Most of the works are focused on a behavioral model for a specific animal in a given environment. Tu and Terzopoulos's work on artificial fish is one example.[19] Blumberg and Galyean proposed an approach that allows an external entity to direct an autonomous creature at multiple levels.[7] Tomlinson and Blumberg studied synthetic social behavior of wild wolves using a combination of hard-coded behaviors and learned behaviors.[18]

Models have been built to mimic animal behavior in crowd simulation.[23] One example for military applications is the US Army's OneSAF Testbed Baseline Semi-Automated Forces (OTBSAF) system[5]. The system has computer-generated forces that can interact with customized training or simulation applications through a connection to a local instance of the Run-Time Infrastructure (RTI) through a gateway. We have developed augmented reality training applications that can communicate with it. The users are reflected in real time in OTBSAF as friendly forces, and the computer-generated forces respond appropriately. These responses are sent to the training system to control the visualizations of the computer-generated forces.[8]

1.2 Animal Animation

With motion capture being widely used in animation, human (and humanoid) character animation has reached high fidelity in motion pictures. It is much more difficult to capture animal motions using motion capture techniques, both because of the difficulties on the equipment side as well as on the cooperation of the animals.

Although animal animations have made much progress, more research is still needed for realistic animation of animal characters due to the vast variety of animal species and different application purposes.

This study is focused on animal behavior and animation for VR applications. Synthetic animal characters are included in the simulation and training system so that their behavior can be programmed and controlled digitally. One example of such an application would be a system that can be used to train how to behave in a manner compatible with cultural norms.

[*] e-mail: zhuming.ai@nrl.navy.mil

[†] e-mail: mark.livingston@nrl.navy.mil

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **MAR 2012**	2. REPORT TYPE	3. DATES COVERED **00-00-2012 to 00-00-2012**

4. TITLE AND SUBTITLE **Configurable Semi-Autonomic Animated Animal Characters in Interactive Virtual Reality Applications**	5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Naval Research Laboratory,4555 Overlook Ave., SW,Washington,DC,20375**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S) 11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

5th Workshop on Software Engineering and Architectures for Realtime Interactive Systems (SEARIS 2012), IEEE Virtual Reality Conference, Orange County, CA, 4-8 Mar 2012.

14. ABSTRACT

In many virtual reality (VR) simulation and training applications it is desirable and even critical to have computer controlled animal characters that can behave with a high degree of realism. The realism can be measured in two aspects. One is behavioral realism or how real the characters act and respond to the commands and environments. The other aspect is visual realism. Many efforts have been directed at animating human characters as well as other animal characters in applications such as interactive computer games. However, more research is still needed for realistic animation of animal characters due to the vast variety of animal species and different application purposes. In some of the environments for which security personnel are being trained, animals, such as dogs, are part of the training programs. So we need to include realistic behavior and visual representations of animal characters in our training applications. This study is focused on animal behavior and animation for VR applications. A simple implementation of real-time animal animation method that is configurable is proposed to make it easy for user interaction. Synthetic animal characters are included in the system so that their behavior can be programmed and controlled digitally. A two stage state machine is used in the system. The top level state machine controls the animal behavior. The secondary state machine controls animation. Animation blending and procedure control are used to make the animation smooth.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**6**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

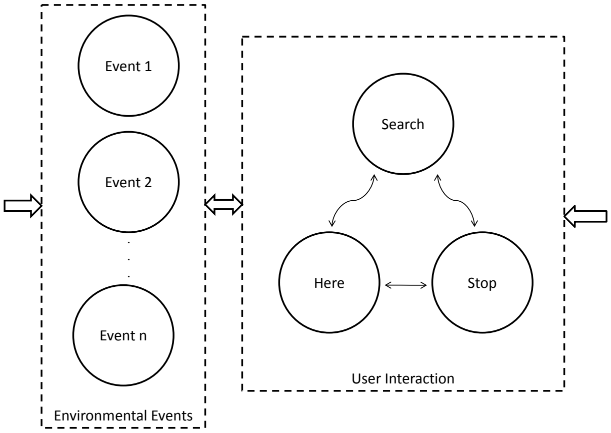

Figure 1: Behavior state machine. Example events in the environment would include goals such as avoiding moving vehicles or interesting conditions, such as a smell that the dog decides to investigate.

We propose in this paper a simple implementation of a real-time animal animation method that is configurable to make it easy for user interaction.

The rest of the paper is organized as follows. In Section 2, a semi-autonomic animal behavior module using artificial intelligence will be proposed. Animal animation will be discussed in Section 3. Then the implementation of the VR system will be presented in Section 4. Discussions of future work will be shown in Section 5. Finally, we will draw conclusions.

2 Animal Behavior

In our applications, we are more interested in well-trained animals, such as dogs, which are directed by a user. Most of the time, the animal is doing what the user asks it to do, but it will also respond to the environment.

The behavior of the synthetic animal characters in the training and simulation VR application is controlled by an AI module. It can be further programmed with a scripting language.

The AI module is a two-stage state machine. The top-level state machine works at mission and animal behavior level (Figure 1). It controls how a character acts when it receives a user command. It also controls the response of the animal when it encounters certain environmental events. For example, when a dog actor encounters an odor, the state machine will send it toward the direction of the odor source. Another example would be a short-term goal of avoiding a moving vehicle, perhaps by simply waiting for it to pass. This two-layered state machine is easily configurable. Animal behavior can be described or modified through configuration files.

2.1 User Interaction

In VR applications involving dogs or other animals, a user should be able to interact with them as in everyday life. In such scenarios, the user may try to command a dog to approach or to go away. And the animal may obey or follow such a request.

A few different user interfaces are designed to input such user commands to the animal. A mouse-and-keyboard interface is the simplest form, and it has been implemented in the system. Other forms, such as gesture recognition using commercially available game controllers, such as Microsoft's Kinect for Xbox 360, are under investigation. Voice recognition is another option that will be studied. All these methods will be combined in the future to form an immersive VR system.

There are many well-trained dogs that are widely used in search and rescue, guide, and security applications. We use a dog as an example in this paper to present the user-animal interaction in the VR application.

Among the common dog commands[3], the following selected commands are implemented in the application.

- Sit
- Down
- Stay
- Stand
- Watch
- Come
- Here
- Search
- Track
- Out
- Go Out

To simplify the implementation of the state machine, these commands are combined into three states as shown on the right side of Figure 1. "Sit", "Down", "Stay", "Stand", and "Watch" are combined as "Stop" state. "Come" and "Here" are combined as "Here" state. "Search", "Track", "Out", and "Go Out" are combined as "Search" state.

When the user issues a command, the state machine will enter one of the user interaction states. In each of the three user command states ("Search", "Here", and "Stop"), a secondary state machine is created to control the animation (this will be discussed in Section 3). A subsequent user command will transfer the state machine to a different state. In each state, extra steps for smooth transitions are carried out, which will also be discussed in Section 3.

These commands may include requested directions. A Search command, for example, will generally include a directional cue that indicates the direction that the user desires for the dog to go. Similarly, the user may indicate a Here state with a specific location relative to the user (e.g. left side or right side) where the dog should sit. These secondary goals are included as a field in the state information. Again, the dog may or may not follow these goals precisely.

2.2 Animal Reaction to Environmental Events

There are many different environmental events that may interrupt what the animal is doing in response to user commands. For example, a certain smell can draw a dog's attention, divert it to a different direction, and cause it to behave completely differently. The following environmental events have been identified for dogs and are being studied:

- The dog is too close to an obstacle, e.g. a building.
- The dog is disturbed by a moving person, object, or other "interesting" things.
- The dog smells something.
- The dog's running direction may change due to the hill incline or cross wind.
- The dog's intent to remain on a road even if the road curves.
- Other events.

Each of these events will send the state machine into a corresponding state. The current implementation has a limited number of states for environmental events. It can be expanded when more detailed animal behavior is needed in the VR application.

There is a possibility that the animal will ignore the environmental events, or even the user's commands if it is in a "bad" mood. In our application, it is assumed that the animal is accustomed to humans and will always follow the orders. So user commands will always transfer the state machine to user interaction states. However, an easy extension is to have a probabilistic selection of whether to follow or ignore a user command. Among all the environmental events that have been identified, the obstacle event (the first one in the list) is different than the rest. The animal can not possibility go through obstacles such as buildings. But it may choose to ignore the other environmental events. Based on the assumption that the animal's attention is easily distracted, environmental events will also always transfer the state to one of the environmental event states in this application. Again, this will eventually be extended to a probabilistic selection of a course of action.

It should be pointed out that the two stages of the state machines are loosely coupled. It is possible to replace the top level state machine with other behavior modules. This flexibility is very useful in research when different algorithms are tested.

3 Animation

The animations of the animal characters are controlled by the second-stage state machine as described above. The state machine determines which animation sequences need to be played. Animation blending and procedural control make sure that the animation sequence transition is smooth.

3.1 Animation State Machine

In each state in the top-level animal behavior state machine, there is a secondary state machine. This secondary state machine controls the low-level animation sequence of the character. For example, it controls the animal's transformation from standing to running, etc.

The following states have been identified as animation states for a dog:

- Stand
- Sit
- Lie down
- Turn
- Run

In each state, there may exist a few parameters that are associated with the state. For example, in the "Turn" state, a vector parameter describes the direction it is turning to. This direction could either be indicated by the user, or determined in the upper level state machine based on the environmental events. In the "Run" state, a speed parameter describes how fast it moves, and further determines if the animal is walking, trotting, or running.

Obviously this list is not complete, but it is sufficient for our current application. For different applications and purposes, more states will be needed. The system is implemented so that new states can be easily added through a configuration file. Although there is still some coding needed in some cases, this structure makes the application easy to expand and improve.

To expand the state machine, the following information is needed. Firstly, if a new animation sequence is necessary, this information should be obtained through motion capture or other methods. Secondly, the state transition table should be modified through a configuration file. And finally, in some cases, some code needs to be added/modified to handle special cases such as multiple paths in the transition table. This will be explained in the next example.

These states are shared by all the animation state machines, but the state transition tables are different. As an example, the animation state machine for "Here," "Search," and "Stop" user interaction states in the behavior state machine is shown in Figure 2. The circles labeled "Sit," "Stand," "Turn," and "Run" are states in the animation state machine ("Lie down" state is not shown in this Figure). The arrowed lines are actions taken in the top level behavior state machine. For example, the arrowed lines labeled "Search" means that they are actions in the "Search" state of the behavior state machine.

These animation state machines not only transfer from one state to the next, as standard finite-state machines do, but they also have the mechanism to control which path to take in case there are multiple possibilities. For example, if the system received a "Search" command, it may start from a "Sit" state, go through "Stand," "Turn," and end at the "Run" state. The small circle in the "Run" state in Figure 2 denotes that the state machine stops here. However, if the user command is "Here," the animation could start from the "Run" state, go to "Turn," and back to the "Run" state. Then it could go to the "Stand" state (dash line in the "Run" state in Figure 2) and finally end at the "Sit" state. In the "Run" state, these two commands take different paths.

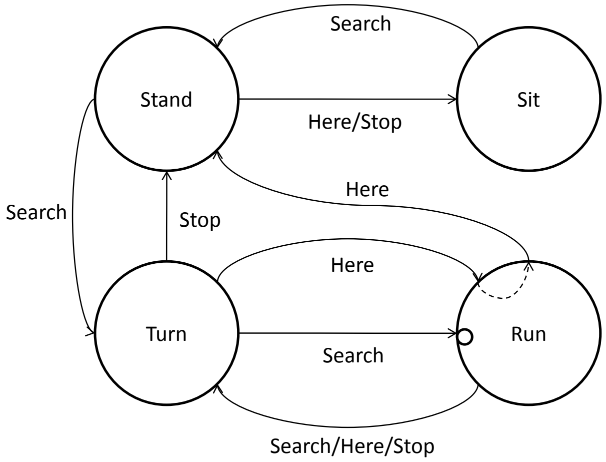

Figure 2: The Animation state machine has states that correspond to animation sequences for the character. Transitions between these states are governed in part by top-level actions; arrows are labeled with the top-level states in which these transitions might occur.

When the state machine enters a state, a pre-defined animation sequence is played. This animation sequence is a smoothed connection of several animation steps. Take the "Run" state as an example, the animation involves walk, trot, and run, depending on the speed the animal is traveling at the time. These animation steps are smoothly connected (which will be discussed in the following) to form a "Run."

3.2 Modeling

We purchased a dog model and a set of animations[1]. The model is modified and converted to the format that meets our needs.

An animated character contains four properties: skeleton, skin mesh, material (including texture mapping), and animation. For a given character, only the animation property changes during the simulation, all three other properties can remain the same.

By some simple modifications to the texture map or skin mesh (e.g. by changing the color of the texture map), we can created multiple animal characters to populate the VR environment with animal crowds. Figure 3 shows two dogs with different colors and slightly different skin meshes.

The available animation sequences are very limited; we extended the animation sequences by editing existing animal poses using 3D character animation software such as MotionBuilder and 3ds Max. Figure 4 shows the dog in the relaxed lying-down pose and the intense lying-down position after modification.

3.3 Animation Blending

It is more difficult to get motion-captured animation sequences for animals than for humans, because animals are not always cooperative. Efforts have been made to use human actors to perform animal actions. Animal animation developed with animation software can also be used. There is also research on animating from video[11, 13], or even from a single picture or a sparse set of pictures of animals[24]. Nonetheless, motion capture is labor-intensive

[1] Purchased from Mixamo, Inc. (http://www.mixamo.com).

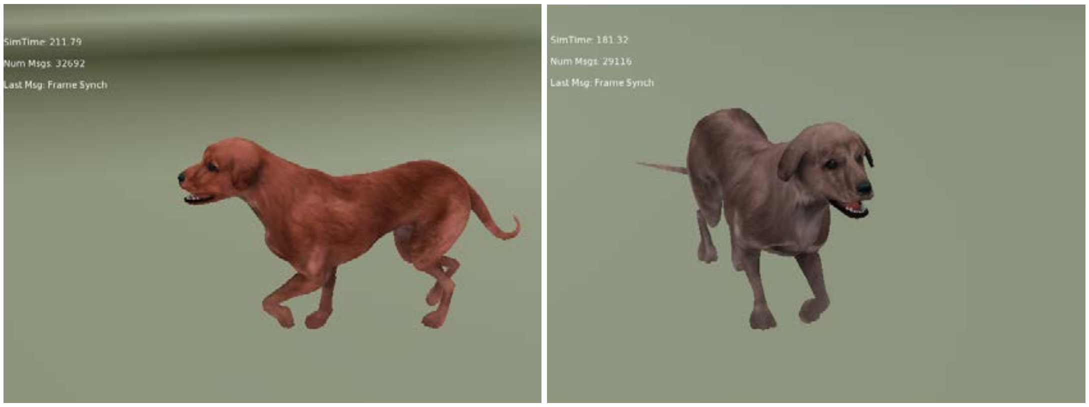

Figure 3: Different variations of animals created by varying the texture properties.

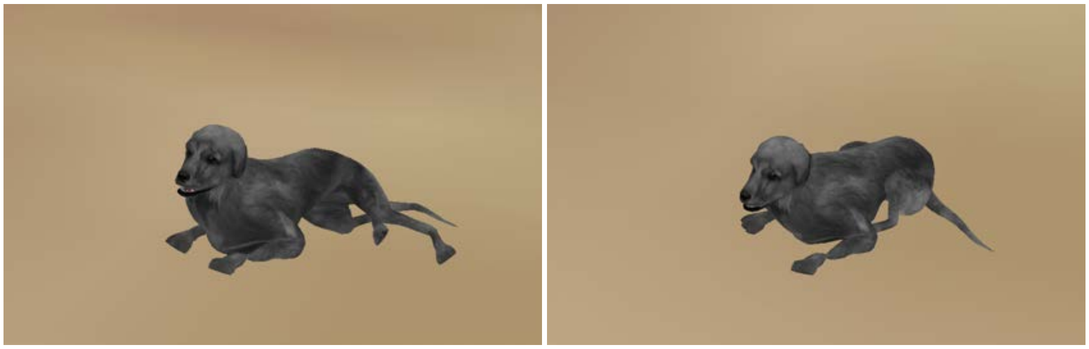

Figure 4: Modifying existing animal pose to expand animation sequence set: a relaxed lying-down pose (left) and an intense version of this pose (right).

and expensive. With a limited number of animation sequences available, animation blending can be used to fill the gaps.

Returning to our example of a dog, a set of animation sequences has been determined that can cover the majority of the movement the dog may perform. These include:

- Run
- Trot
- Walk
- Walk happily
- Lay down
- Sniff
- Idle
- Sit

These sequences are connected with interpolations. Cal3D is a 3D character animation library based on skeletal animation.[1] It supports combining animations and actions. This development toolkit is used in our project to build smooth animal animations. Take the "Run" state in the animation state machine as an example. It starts with an "Idle" animation and is then blended with "Walk." When the speed goes up, it is further blended with "Trot," and finally blending into "Run."

It needs to be pointed out that animation blending works at a level beneath the animation state machine. It can either make the transition between the states smooth, or combine several animation sequences together into one action that otherwise may need a few states in the state machine to implement. The state machine and animation blending work together to accomplish smooth animation.

Figure 5 shows a sequence from a dog animation. The dog was sitting at the beginning, waiting for a command. When it received a "Search" command, it stood up, turned, and started to run with accelerating speed. This sequence is controlled by a series of state changes in the state machine. The changes between states are further smoothed by animation blending.

In some cases, an animation sequence could be interrupted by another animation before it is finished. This has been taken care of on both the state machine level and the animation engine level. The state machine will make sure a proper transitional animation sequence is inserted, and the animation engine will blend the animation to make the transition smooth.

3.4 Real-Time Control

Not all animal movements can be pre-recorded with motion capture techniques, simply because there are an unlimited number of possible movements. There are also aspects of the animation that can not be pre-determined. For example, the animal may turn its head to a certain object that draws its attention. How much the head turns can not be pre-determined and simulated with a limited pool of animation sequences. Although many movements can be approximated with a limited number of animation sequences, it is not desirable due to poor aesthetic quality. It is important to control some aspects of the animation through software procedures.

One common method used to control animation is inverse kinematics.[20] However, this method is not suitable for real-time VR applications. Guerrero et al. developed a blended inverse kinematics method to achieve real-time animation blending.[12] This method is used in our system to control the animal in real-time.

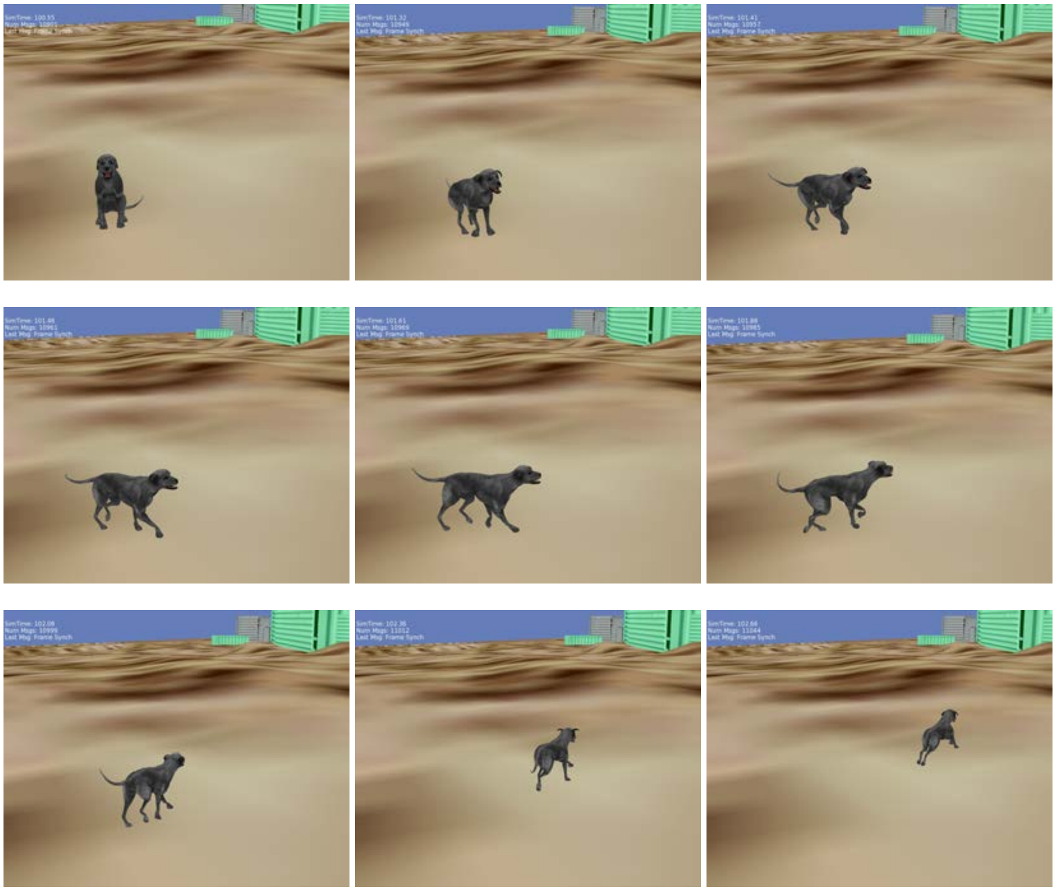

Figure 5: Animation sequence of the “Search” command. Top left: dog in the “Stop” state in the top-level state machine and in the “Sit” state in the animation state machine. Top center: transitioning to a “Search” state in the top-level machine requires a transition in the animation machine to a “Stand” state. Top right: the animation state machine has transitioned to a “Turn” state to orient the dog properly for the “Search” state in accordance with the user command to search in the direction away from the user’s position. Middle row: the dog has reached the “Run” state in the animation machine. Bottom row: the dog accelerates between the left and right images. Animation blending makes the acceleration smooth.

4 Implementation

The VR application software system is implemented around an open source game engine, Delta3D[2], which uses an OpenGL based open source toolkit, OpenSceneGraph[4], for rendering. An open source approach has been implemented to include animated characters into the system. It uses a 3D character animation library, Cal3D.[1]

Our VR application can reach a real-time frame rate without any problem. On a desktop system with a 2.4GHz Intel Core 2 CPU, 3GB of memory, and an Nvidia GeForce 8800 GTX graphics system, the frame rate is about 37 frames/second with a moderately complex environment model.

The system has a modular design. The AI module is separated from the rendering module. A network layer has been designed to exchange animal character status and user status information with external modules. This is useful when more sophisticated animal behavior modules and environmental models are developed separately.

State machines on both stages can be easily reconfigured through configuration files.

5 Discussion

One of the issues that interests us is to integrate the animal animation into an existing simulation system. There is heavy investment and wide usage of some video game-like simulation and training environments that have many more features than our Delta3D based system. For example, Virtual Battlespace 2 (VBS2) is a battlefield simulation system that is used by the U.S. Marine Corps, Australian Defense Force, and many other military users.[16] For systems using standard interfaces, such as high level architecture (HLA)[10], it might be feasible to integrate a customized animated character into such a system. Otherwise, the possibility of such an integration largely depends on how open the existing system is.

For example, the dog model in the current version of VBS2 has a very small set of animation sequences, and the animation and behavior are not adequate for realistic training and simulation applications. Our VR system is designed in a modular fashion. This has the potential to be implemented as a “plug-in” or a module that can be integrated into existing simulation systems. The initial study shows that, although there are many obstacles, this approach is feasible. So far we have successfully imported the animated dog model into VBS2. With the help of the development tools such as VBS2 Fusion, it is possible to control the animal behavior with an external module. It is also possible to control the animation, may be blended with the animation engine, with external modules. We will further investigate such an issue in the near future.

Rendering large crowds of animal characters is another issue that needs more investigation. How many characters can a computer system handle given all the detailed animation and behavior that is involved? Although it is not the focus of our work at this time, it is a subject that is worth further study.

6 Conclusions

In some of the environments for which security personnel are being trained, animals, such as dogs, are part of the training programs. Therefore, it is important to include realistic behavior and visual representations of animal characters in training and simulation applications. In this paper, we proposed a schema for developing animated animal characters in interactive VR applications. It is a configurable two-stage state machine based animal behavior simulation and animation. The top level state machine controls the animal behavior. The secondary state machine controls animation. Animation blending and procedure control are used to make the animation smooth.

This study is concentrated on well-trained animals such as dogs. These animals are under the direction of the user most of the time, but respond to the environment as well. The emphasis has been put on making the behavior easily reconfigurable. The two-state state machine structure works very well for this purpose.

References

[1] Cal3D - 3d character animation library. http://gna.org/projects/cal3d/.

[2] Delta3D. http://www.delta3d.org/.

[3] Dog Commands Language Table. http://www.searchk9.net/pages/training/commands.htm.

[4] OpenSceneGraph. http://www.openscenegraph.org/projects/osg.

[5] R. D. Anderson and M. V. Baranoski. OneSAF Test Bed (OTBSAF) Automation. Technical report, ARMY RESEARCH LAB ABERDEEN PROVING GROUND MD WEAPONS AND MATERIALS RESEARCH DIRECTORATE, May 2005.

[6] R. Aylett and M. Luck. Applying Artificial Intelligence to Virtual Reality: Intelligent Virtual Environments. *Applied Artificial Intelligence*, 14(1):3–32, 2000.

[7] B. M. Blumberg and T. A. Galyean. Multi-level direction of autonomous creatures for real-time virtual environments. In *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 47–54, New York, NY, USA, 1995. ACM.

[8] D. G. Brown, J. T. Coyne, and R. Stripling. Augmented Reality for Urban Skills Training. In *VR '06: Proceedings of the IEEE conference on Virtual Reality*, pages 249–252, Washington, DC, USA, 2006. IEEE Computer Society.

[9] P. A. G. Calero and M. A. Gómez-Martín. *Artificial Intelligence for Computer Games*. Springer, 2011.

[10] J. S. Dahmann. High Level Architecture for Simulation. *Distributed Interactive Simulation and Real-Time Applications, International Workshop on*, 0:9+, 1997.

[11] L. Favreau, L. Reveret, C. Depraz, and M. P. Cani. Animal gaits from video. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, SCA '04, pages 277–286, Aire-la-Ville, Switzerland, Switzerland, 2004. Eurographics Association.

[12] M. Guerrero and C. Darken. Blended Inverse Kinematics: Delta3D System Utilization. *The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC)*, 2008(-1), Jan. 2008.

[13] S. L. Hannuna, N. W. Campbell, and D. P. Gibson. Identifying quadruped gait in wildlife video. In *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, volume 1, pages I–713–16. IEEE, 2005.

[14] J. Li and D. Huang. Modeling and Animation of Realistic 3D Virtual Human Using Inertial-sensor. *Optoelectronics and Image Processing, International Conference on*, 1:712–715, 2010.

[15] H. Orozco, F. Ramos, M. Ramos, and D. Thalmann. An action selection process to simulate the human behavior invirtual humans with real personality. *The Visual Computer*, 27(4):275–285, Apr. 2011.

[16] M. J. Singer and J. Barnett. Military training in virtual environments: User interface evaluations. In *Human Factors and Ergonomics Society Annual Meeting Proceedings, Virtual Environments*, pages 2403–2407. Human Factors and Ergonomics Society, 2010.

[17] L. Skrba, L. Reveret, F. Hétroy, and M.-p. Cani. Quadruped Animation. In T. Theoharis and P. Dutré, editors, *Eurographics State-of-the-Art Report*, pages 1–17. 2008.

[18] B. Tomlinson and B. Blumberg. AlphaWolf: Social Learning, Emotion and Development in Autonomous Virtual Agents. In *Proceedings of First GSFC/JPL Workshop on Radical Agent Concepts*, pages 35–45, 2002.

[19] X. Tu and D. Terzopoulos. Artificial fishes: physics, locomotion, perception, behavior. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, SIGGRAPH '94, pages 43–50, New York, NY, USA, 1994. ACM.

[20] L. Unzueta, M. Peinado, R. Boulic, and A. Suescun. Full-body performance animation with Sequential Inverse Kinematics. *Graphical Models*, 70(5):87–104, Sept. 2008.

[21] K. Wampler and Z. Popović. Optimal gait and form for animal locomotion. *ACM Trans. Graph.*, 28(3):1–8, July 2009.

[22] S. Wermter, G. Palm, and M. Elshaw, editors. *Biomimetic neural learning for intelligent robots: Intelligent systems, cognitive robotics, and neuroscience*, volume 3575 of *Lecture Notes in Computer Science*. Springer, 1 edition, Aug. 2005.

[23] H. Xiao and C. He. Behavioral Animation Model for Real-Time Crowd Simulation. In *Advanced Computer Control, 2009. ICACC '09. International Conference on*, pages 250–254. IEEE, Jan. 2009.

[24] X. Xu, L. Wan, X. Liu, T. T. Wong, L. Wang, and C. S. Leung. Animating animal motion from still. In *ACM SIGGRAPH Asia 2008 papers*, SIGGRAPH Asia '08, New York, NY, USA, 2008. ACM.